



[12] 发明专利申请公开说明书

[21] 申请号 94193269.9

[51] Int. Cl.⁶

G05B 19/042

[43] 公开日 1996年9月4日

[22] 申请日 94.6.16

[30] 优先权

[32] 93.7.29 [33] US[31] 08 / 098,790

[86] 国际申请 PCT / US94 / 06858 94.6.16

[87] 国际公布 WO95 / 04314 英 95.2.9

[85] 进入国家阶段日期 96.3.4

[71] 申请人 费舍-柔斯芒特系统股份有限公司

地址 美国得克萨斯州

[72] 发明人 特伦斯·L·布莱文斯

[74] 专利代理机构 上海专利商标事务所

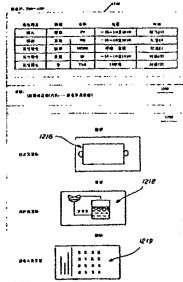
代理人 沈昭坤

权利要求书 8 页 说明书 24 页 附图页数 15 页

[54] 发明名称 统一控制模板生成系统和进程控制程序设计的方法

[57] 摘要

一控制模板，它表示某控制环境下控制进程的一个被选函数并用来生成被选函数的多个可显示的视图。模板包括诸如控制进程的算法和已知控制参数等进程控制信息、诸如输入和输出等进程环境的属性信息、至少一个与进程控制函数信息相关的用以通过被选控制模板视图而使控制进程的具体控制参数相关联的对话集，以及使用属性信息和至少一个对话集来形成进程控制解的控制方法指令。还揭示了一种用于生成或修改控制模板的系统。



权 利 要 求 书

1. 一种进程控制系统,其特征在于,它具有控制模板,每个模板代表某控制环境下控制进程的一个被选函数,并且在该控制环境中可显示被选控制进程的不同控制模板视图,该系统包括:

A. 中央处理器(116);

B. 可由中央处理器访问的多个控制模板(23、25、26 和 27),其中每个控制模板包括:

(1)进程控制函数信息(132);

(2)进程环境的属性信息(10,602);

(3)至少一个与进程控制函数信息相关的对话集(42, 608),其通过被选的控制模板视图联系控制进程的具体控制参数;和

(4)控制方法指令(700),供使用属性信息和至少一个对话集形成进程控制解;以及

C. 与中央处理器连接的用户接口设备,它能使用户选择控制模板的所需视图,以作显示,并能使用户访问属性信息和相关的对话集,以使用户按需生成和/或编辑被选的控制模板。

2. 如权利要求 1 所述的进程控制系统,其特征在于,每个所述控制模板都具有如技术员视图、操作员视图、控制员视图和维修视图之类的多个被选函数的可选控制模板视图(34、36、38 和 40)。

3. 如权利要求 1 所述的进程控制系统,其特征在于,每个所述的控制模板具有多个与控制函数信息相关的对话集。

4. 如权利要求 1 所述的进程控制系统,其特征在于,可用 Fortran 和 C 之类的各种程序设计语言来生成控制方法指令。

5. 如权利要求 1 所述的进程控制系统,其特征在于,它还包括控制模板库(123),用于存储表示多个控制模板(130)的数据。

6. 如权利要求 5 所述的进程控制系统,其特征在于,控制模板库进一步包括:

表示包括预先定义控制模板的第一组控制模板的数据;和

表示用户生成的第二组控制模板的数据。

7. 如权利要求 1 所述的进程控制系统,其特征在于,还包括模板生成器(124),它与中央处理器和用户接口设备互连,允许用户通过访问和改变代表进程控制函数的数据、属性信息、相关对话集、控制方法或控制模板视图来编辑现有的控制模板或生成新的控制模板。

8. 如权利要求 7 所述的进程控制系统,其特征在于,所述模板生成器包括图形生成器(128),它允许用户为被生成或被编辑的控制模板设计与被选函数相关的图形视图。

9. 如权利要求 1 所述的进程控制系统,其特征在于,用户可以选择控制方法的所述指令。

10. 如权利要求 1 所述的进程控制系统,其特征在于,还包括:

形成用户接口设备(118)一部分的显示设备;和

模板生成器界面视屏(200),该视屏被显示在显示设备上并有多个视屏提示符供用户选择,所述用户界面设备(118)可使用户访问属性信息和控制方法指令,以便编辑或生成新的控制模板(130)。

11. 如权利要求 1 所述的系统,其特征在于,还包括被保存的设备数据,用户可用用户接口设备访问所述数据,以显示与被选控制模板相关的设备的实际结构。

12. 如权利要求 11 所述的系统,其特征在于,还包括显示装置,以允许用户通过用户接口设备监视设备的工作情况。

13. 如权利要求 1 所述的系统,其特征在于,进程控制信息包括诸如控制进程的算法和/或已知控制参数等信息。

14. 如权利要求 1 所述的系统,其特征在于,属性信息包括诸如

进程输入和/或输出的信息。

15. 一种控制模板,其特征在于,它表示某控制环境下控制进程的一个被选函数,该模板用来生成被选函数的多个可显示视图,诸如技术员视图、操作员视图、控制员视图、维修视图和类似视图,所述模板包括:

进程控制函数信息(132);

进程环境的属性信息(10,602);

至少一个与进程控制函数信息相关的对话集(42),用于通过被选的控制模板视图联系控制进程的具体控制参数;和

控制方法指令,用于使用属性信息和至少一个对话集来形成进程控制的解。

16. 如权利要求 15 所述的控制模板,其特征在于,还包括被选函数的多个可选择的控制模板视图(34、36 和 38)。

17. 如权利要求 16 所述的控制模板,其特征在于,所述至少一个对话集通过被选的控制模板视图与进程控制函数的控制参数关联。

18. 如权利要求 17 所述的控制模板,其特征在于,控制模板视图映出进程控制技术员工视图、进程控制操作员视图、实验人员视图、维修人员视图或类似的视图。

19. 如权利要求 15 所述的控制模板,其特征在于,还包括与进程控制函数信息相关的多个对话集。

20. 如权利要求 15 所述的控制模板,其特征在于,进程控制函数信息包括诸如控制进程的算法和已知控制参数的信息。

21. 如权利要求 15 所述的控制模板,其特征在于,属性信息包括诸如进程输入和输出的信息。

22. 一种为进程控制系统生成或编辑控制模板的方法,其特征在于,每个模板表示某控制环境下控制进程的一个被选函数,在该控

制环境中可以显示一个被选控制进程函数的不同控制模板视图,所述方法包括下列步骤:

A. 通过中央处理器访问多个控制模板(23、25、26 和 27),其中每个控制模板包括:

(1)进程控制函数信息(132);

(2)进程环境的属性信息(10,602);

(3)至少一个与进程控制函数信息相关的对话集(42, 608),其通过被选的控制模板视图联系控制进程的具体控制参数;和

(4)控制方法指令(700),其用属性信息和至少一个对话集形成进程控制解;

B. 使用户接口设备(118)与中央处理器相连;

C. 使用户通过用户接口设备选择控制模板的所需视图,以作显示;并且

D. 访问属性信息和所述的至少一个相关对话集,以使用户按需生成和/或编辑被选的控制模板。

23. 如权利要求 22 所述的方法,其特征在于,还包括下述步骤,即为每个所述控制模板的被选函数提供技术员视图、操作员视图、控制员视图和维修视图之类多个可选择的控制模板视图(34、36、38 和 40)。

24. 如权利要求 22 所述的方法,其特征在于,还包括对于每个所述控制模板,把多个对话集与控制函数信息联系起来的步骤。

25. 如权利要求 22 所述的方法,其特征在于,还包括用诸如 Fortran 和 C 等各种程序设计语言生成所述控制方法指令的步骤。

26. 如权利要求 22 所述的方法,其特征在于,还包括将表示多个控制模板(130)的数据存储在控制模板库(123)中的步骤。

27. 如权利要求 26 所述的方法,其特征在于,将表示多个控制模板(130)的数据存储在控制模板库(123)中的步骤进一步包括下列

步骤:

存储表示第一组控制模板的数据,第一组控制模板包括预先定义的控制模板;和

存储表示用户生成的第二组控制模板的数据。

28. 如权利要求 22 所述的方法,其特征在于,还包括下列步骤:
提供与中央处理器和用户接口设备互连的模板生成器(124);和
通过访问和改变表示进程控制函数的数据、属性信息、相关对话集、控制方法或一个控制模板视图,用模板生成器编辑现有控制模板或生成新的控制模板。

29. 如权利要求 28 所述的方法,其特征在于,还包括下列步骤:
使模板生成器包括一图形生成器(128);和
用图形生成器为被生成或被编辑的控制模板设计与被选函数相关的图形视图。

30. 如权利要求 22 所述的方法,其特征在于,还包括选择所述控制方法指令中的预定指令,以形成进程控制解的步骤。

31. 如权利要求 22 所述的方法,其特征在于,还包括下列步骤:
包括一作为用户接口设备(118)一部分的显示设备;
提供模板生成器界面视屏(200),该视屏将被显示在显示设备上并具有多个视屏提示符可供用户选择;和

用所述用户接口设备(118)访问属性信息和控制方法指令,以便编辑或生成新的控制模板(130)。

32. 如权利要求 22 所述的方法,其特征在于,还包括下列步骤:
存储设备数据;和
用用户接口设备访问被存的设备数据,以显示与被选控制模板相关的设备的实际结构。

33. 如权利要求 22 所述的方法,其特征在于,还包括用用户接口显示设备监视设备工作情况的步骤。

34. 如权利要求 22 所述的方法,其特征在于,还包括使进程控制信息中包括诸如控制进程的算法和/或已知控制参数等信息的步骤。

35. 如权利要求 22 所述的方法,其特征在于,还包括下述步骤,即包括诸如进程输入和/或输出等信息作为属性信息。

36. 一种用表示某控制环境下控制进程的一个被选函数的控制模板来生成进程控制解的方法,其特征在于,所述方法用控制模板生成技术员视图、操作员视图、控制员视图、维修视图之类被选函数的多个可显示视图,所述方法包括下列步骤:

提供进程控制函数信息(132);

提供进程环境的属性信息(10,602);

至少提供一个与进程控制函数信息相关的对话集(42),其通过被选的控制模板视图联系控制进程的具体控制参数;和

依靠属性信息和至少一个对话集,用控制方法指令形成进程控制解。

37. 如权利要求 15 所述的方法,其特征在于,还包括使用控制模板的步骤,所述控制模板具有被选函数的多个可选择的控制模板视图(34、36、38)。

38. 如权利要求 37 的方法,其特征在于,还包括通过被选的控制模板视图使所述的至少一个对话集与进程控制函数的控制参数联系起来步骤。

39. 如权利要求 38 的方法,其特征在于,还包括用控制模板视图来映出进程控制技术员工视图、进程控制操作员视图、实验人员视图、维修人员视图或类似视图的步骤。

40. 如权利要求 36 的方法,其特征在于,还包括将多个对话集与进程控制函数信息联系起来的步骤。

41. 如权利要求 38 的方法,其特征在于,还包括在进程控制函

数信息中包括诸如控制进程的算法和已知控制参数等信息的步骤。

42. 如权利要求 38 的方法,其特征在于,还包括下述步骤,即包括诸如进程输入和/输出等信息作为属性信息。

说 明 书

统一控制模板生成系统和进程控制程序设计的方法

著作权通告

本专利文件的部分公开内容包含受著作权保护的材料。著作权人不反对任何人对专利和商标局文档或记录中出现的专利文件或专利公开内容作传真复制,但在别的方面却保留任何所有的著作权。

发明领域

本发明总体上涉及进程监视和控制系统,尤其涉及一种用于生成控制模板的系统,其中控制模板具有若干属性、方法和与其相关的图形视图,用户可对此作选择以生成设计进程控制的解,并由此产生技术员视图、操作员视图和控制员视图之类的独特显示视图。技术员视图中属性和方法的任何变化将自动反应在使用这些属性和方法的所有其他视图中。

背景技术

进程控制包括使用仪表、控制设备以及用于测量和操纵诸如阀门等控制元件的系统,以便使温度、压力和流量等一个或多个进程变量保持在所选的目标值,获得工艺进程的预期目的,包括进程中使用的机器和设备的安全和有效运行等。进程控制系统在自动化的生产进程中具有广泛的应用,例如用于化工、石油和制造工业等方面。

通常是用使用微处理器的控制器或计算机来控制进程的,它们通过向硬件设备发送并接收指令和数据来监视进程,从而控制进程的某一特殊方面或整个进程(作为一个整体)。无需改变硬件,通过程序设计便可逐一设计、改进或修改这些微处理器或计算机中软件程序所完成的特定进程控制函数。例如,技术员可编写一段程序,该程

序使控制器(通过液面传感器)读出储罐中的液面高度,将储罐中的液面高度与预先定义的希望高度作比较,然后根据所读高度是否低于或高于预先定义的希望高度打开或关闭送料阀。通过打开显示进程的技术员视图并使用它来修改程序,可以方便地改变参数。

除了执行控制进程外,软件程序还可用于以操作员显示或视图的形式提供有关特定进程状态的反馈,当发生故障时发出报警信号,或者当发生故障时为操作员提供指令或建议。负责控制进程的操作员需从他的角度观察该进程。显示器或控制台通常是执行进程控制函数的使用微处理器的控制器或计算机与操作员之间的界面,并且也是程序员或技术员与执行进程控制函数的使用微处理器的控制器或计算机之间的界面。

在进程控制环境中运行、监视、控制和反馈函数的系统一般是用诸如 Basic、Fortran 或 C 等高级计算机程序设计语言编写并在计算机或控制器上执行的软件来实现的。尽管这些高级语言对进程控制的程序设计很有效,但通常不为进程技术员、维修技术员、控制技术员、操作员和管理员所使用或理解。对于这类人员,已开发了更高级的图形显示语言,例如连续函数块和梯形逻辑(ladder logic)等。这样,每个技术员、维修人员、操作员、试验人员和其他人员都需要一个该进程控制系统各元件的图形视图,以便他们能够观察系统中与其责任相关的部分。

例如,进程控制程序可以用 Fortran 语言编写,并且它需要两个输入数据,计算输入数据的平均值并产生一个等于这两个输入数据之平均值的输出值。该程序可称为平均函数,而且控制技术员可通过图形显示调用和引用它。典型的图形显示可以由一矩形方块组成,它具有两个输入端、一个输出端和一个表示该块为平均函数的标记。可用不同的程序产生同一函数的图形表示以供操作员观察平均值。在把该系统传递给用户之前,先把这些软件程序放入一具有用户预先

选定属性的库中。各种程序是由函数块来识别的。于是用户可以调用函数并选择预先定义的图形表示,为操作员、技术员等产生不同的视图,例如其方法可以是所述库中的多个函数块中选择一个函数块来确定进程控制解,不必再用 Fortran 语言建立一个全新的程序。

可把一组每个都由一相关的函数块指定的标准化函数存储在一控制库里。拥有这种库的设计者能通过在一计算机显示屏上互连各种通过函数块所选的函数或元件来设计进程控制解,以完成特定的任务。微处理器或计算机把每个函数块定义的函数或元件与存储在库中的预先定义模板联系起来,并且根据设计者所希望的互连使每个程序函数或元件相互关联。理想地,设计者甚至不必用 Fortran 或其他高级的程序设计语言便能用预先定义函数的图形视图设计整个控制程序。

用图形视图进行进程控制程序设计的一个相关问题是,现有的系统只允许设备制造商而不是该设备的用户,生成他自己的控制函数以及相关的图形视图或修改所提供的程序库内预先定义的函数。

因此,新的进程控制函数主要是由销售设计系统的公司而不是最终用户设计的,这些用户可能对该公司提供的标准函数组以外的函数有特殊的需要。标准函数包含在一控制库中,该控制库提供面向最终用户的系统。最终用户要么使用设计环境提供的现有函数,要么依靠提供该设计环境的公司为其开发任何希望特别定制的函数。如果要求设计者改变技术员视图的参数,那么必须相应地重写和修改使用这些参数的所有其他视图,因为函数程序和视图程序通常是独立开发的并且不是某个综合开发环境的一部分。显然,该进程非常麻烦,昂贵并且费时。

人们需要一种统一的或普适的设计环境,不仅设计员或制造商而且还有用户都能方便地用它改制一现有的解来满足其开发进程控制函数的特殊需要。这种设计环境使进程控制设计员或用户能够修

改标准进程控制函数或生成特别定制的进程控制函数,并生成与修改后或新生成的进程控制函数相关的图形视图,所有这些都在一公共的环境中进行。该设计环境应该为函数的生成和有关的技术员、操作员、试验和维修人员或其他可能的用户提供一个公共界面,以便在修改或生成技术员的函数后,该修改或生成会在该函数所有的图形视图中显现出来。另外,该设计环境应该具有一个包含与进程控制函数相关的属性、方法和图形的公共数据库结构,以允许用设计员所希望或要求的任何图形方法来表示经修改的或生成的进程控制函数,与其他希望的图形视图一样,不管是用梯形逻辑、连续函数块还是各类技术员、操作员、试验员和维修人员需要的其他设计语言。

发明内容

本发明的系统解决了上述问题并实现了技术改进,该系统用于设计专门的控制模板,它可定义进程控制程序设计环境中使用的进程控制函数。每个控制模板都同时具有程序化的函数和这些函数的图形视图,从而可以选择某个特定的图形视图。该系统提供了一个统一的设计环境,允许用户开发进程控制函数以及相关的函数块或标识函数各视图的图标,它可以作为控制模板与其他模板一起存储在一公共控制模板库中。

在一实施例中,为设计专门的控制模板提供了一个进程控制程序设计环境,这些专门控制模板用来定义在进程控制环境中设计控制方法时所用的进程控制函数。该环境(通常称为工作站)包括一具有键盘和/或鼠标器之类输入设备的中央处理器和—在用户和中央处理器之间提供可视界面的图形显示器。控制模板生成器用与专门控制模板表示的进程控制函数相关的属性、方法和图形视图的形式,将进程控制函数数据存储起来。用一控制模板库存储新生成的和原先存在的控制模板。

在本发明的另一实施例中,提供了一种控制模板生成器,它包括

一属性和方法语言生成器和一图形生成器,前者用于存储进程控制函数数据,它定义将与表示一特定进程控制函数的控制模板相关联的属性和方法,而后者用于设计与控制模板所表示的进程控制函数相关的图形视图。这样,属性和方法语言生成器和图形生成器为控制模板的生成提供了一个统一的设计环境。属性和方法语言生成器提供了设计界面显示屏,它们允许用户定义或选择所有将与某个特殊进程控制函数相关联的属性,并选择或生成将与该进程控制函数相关联的方法。图形生成器为用户提供了一个可用来生成图形视图的显示屏界面,这些图形视图示出了用户选定的用来定义该进程控制函数的属性和方法。图形生成器以类似与属性和方法语言生成器的方式提供了一个显示屏界面,并允许用户生成将与控制函数相关联的图形视图。只作为举例,这种视图包括技术员视图、操作员视图、控制视图、维修视图和其他需要的视图。一旦用户设计了新的控制模板,控制模板生成器便把代表该新的控制模板的数据存储在控制模板库中。并可在设计、修改和执行进程控制程序和操作员和维修等视图时使用存储在控制模板库内的控制模板函数数据。因此,每个控制模板都包括算法、输入、输出、其他属性和用于执行进程控制函数的程序。另外,有多个对话集(conversation set)与控制模板中定义每个属性相关联。对话集定义了可从一确定视图请求的用户交互,它们与那些特定的进程控制函数相联系。对话集包括对软键、视窗和字段(field)等的定义,使用户能与模板通信并在函数中插入新的属性值或修改旧的属性值。因此,如果修改了与控制模板中定义的某一属性相关的对话限制等,那么所有引用该属性的图形视图都将反映出该变化。

本发明获得的重要技术优势在于,用户可以修改现有的模板或生成他自己的在进程控制中使用的控制模板,而不是完全依赖控制模板库内由数据环境制造商提供的标准模板。如此生成或修改的模

板包括对生成任何所需图形视图所必要的属性定义。从而,用户可从单个经修改的或新生成的控制模板中生成多个常规的图形视图,以跟诸特定的控制函数发生关系,并把所生成的具有控制函数的视图存储起来为以后引用。

附图概述

在后附的权利要求中给出了认为是本发明新特点的特征。结合附图参考以下优选实施例的详细描述将更能透彻地理解本发明本身及其其他目的、特征和优点,其中同样的标号表示同样的元件,并且

图 1, 2 和 3 是描绘已有技术进程控制环境的方框图;

图 4 是本发明工作站的一个综合实施例,它只允许为一个视图(诸如技术员视图)生成新的控制模板或修改现有的控制模板;

图 5A—G 是本发明优选实施例的基本方框图;

图 6—13 是图 4 环境中工作站的逐个视屏显示,描绘了设计员是如何根据本发明生成或修改一控制模板的;

图 14 和 15 是工作站视屏显示,分别示出了构造技术员视图和操作员视图的图形编辑界面;而

图 16 是一张概括控制模板元件的表。

本发明的最佳实施方式

图 1 示出了进程控制环境 100,它描绘了用来实现已有技术的进程控制器的控制环境。进程控制系统 100 包括由局域网(“LAN”)108 电气互连的操作员工作站 102、实验室工作站 104 和工程设计工作站 106,可以向和从各工作站和控制器 110 发送和接收数据和控制信号。图中示出工作站 102、104 和 106 通过 LAN 108 与多个在诸工作站和多个进程 112 之间提供电气接口的控制器/多路复用器 112 相连。可以理解,LAN 108 可根据进程控制环境的需要或只包含一个与控制器/多路复用器 110 直接相连的工作站,或包含许多工作站 102、104、106 和许多控制器/多路复用器 110。也可理解,单个进

程控制器/多路复用器 110 可负责控制几个不同的进程 112, 或只负责控制单个进程的一部分。

例如在该环境中, 通过在工程设计工作站 106 上生成一软件控制解来开发进程控制策略, 并通过 LAN 108 把解传送给操作员工作站 102、实验室工作站 104 和控制器/多路复用器 110 执行。操作员工作站 102 和实验室工作站 104 为控制器/多路复用器 110 中执行的控制/监视策略提供界面显示, 并根据设计解的需要与一个或多个控制器/多路复用器 110 通信, 以便观察进程 112 和改变控制属性值。另外, 如以下将进一步讨论的那样, 操作员工作站 102 和实验室工作站 104 可把与被控进程 112 的状况和状态相关的声像反馈给操作员。工程设计工作站 106 包括中央处理器(CPU)116, 以及显示器和键盘、光笔之类的输入/输出设备或用户接口设备 118。CPU 116 还可以具有它自己的存储器 117。尽管没有示出, 但可以理解, 进程控制环境 100 内的所有工作站都至少包括一个与显示器和用户接口设备 118 电气连接的中央处理器 116, 使用户能和 CPU 交互。

函数块库 122 通常与工程设计工作站 106 的 CPU 116 耦合, 也可以是 CPU 116 的一部分, 并且为进程控制和监视提供固定的函数块。设计员将这些函数块提供给最终用户, 以在特定的进程控制环境 100 中使用。

如图 2 所示, 函数块库 122 的技术员视图包含许多用来定义和访问特定函数的函数块 22、23、24、25、26、27、28、29、30 和 31。如虚线 119 所示, 这些函数块可被 CPU 116 访问, 以在所示的显示器 118 上定义监视/控制策略。因此, 可为温度 T1 和温度 T2 定义代表温度测量函数的平均输入函数块 22(AI), 并且在显示器 118 中选择和包括该函数块。如果需要对这些温度取平均, 则可选择平均函数的函数块 23 并使温度 T1 和 T2 与函数块 23 表示的平均模板耦合。函数块 23 表示的平均函数的输出可以与选自函数块库 122 的函数块 26 表

示的PID函数耦合。函数块26表示的PID函数的输出可接至函数块28表示的且选自函数块库122的输出设备函数。这样,进程控制/监视策略可用函数块库122中的各种模板表示,而模板则由函数块22、23、26和28来选择并于显示器118上耦合在一起,图示出该进程控制/监视策略。然而应该理解,库122中的每个控制模板具有与其相关的固定参数,以使它们只能以固定或预先定义的方式相互耦合。因此,如果想把流量测量联接到表示平均函数的函数块23而不是函数块26和28,则系统设计员必须生成一新的函数块存储在库122中,以便用户以后用类似于图2所示的方式进行观察。

图3示出了已有技术中这种函数块的生成。函数块22表示的新函数125包括进程控制函数123,它具有输入10和输出12,以及与进程控制函数相关的算法14和任何已知类型的参数16和18。固定数据20也与进程控制函数123有关,并且包括固定的滤波系数、固定的时间间隔等数据。生成该函数的技术员视图22以表示已生成的函数块125,并将视图22看作函数块125的一部分。然后存储函数块125,并表示函数块库122中已生成的函数块,使其被访问时弹出已生成的函数块125。于是,可将其加入前述图2中显示器118上所示的监视/控制策略的定义中。

但值得注意的是,函数块125是专门为工程设计视图设计的。因此,对图2中函数块118表示的监视/控制策略的任何修改将不会出现在操作员视图、维修视图、实验室视图、或任何其他视图中。如果该修改对于操作员、控制员、实验人员、维修人员等是必要的,则必须修改每个界面显示,以便访问和显示新函数块的各个属性。换句话说,没有可被引用来修改界面的预先定义视图。这是因为只定义了工程设计视图,而且没有对生成需要的操作员显示,实验室显示或其余视图所需的预先定义视图的支持。

图4是本发明较佳实施例的基本方框图。它包括图1所示已有

技术的所有元件,另外还有模板生成器 124 和控制模板库 123,两者形成控制模板系统 120。在已有技术的类似元件与本发明的类似元件一致之处,使用相同的标号。本发明的控制模板被定义为属性函数的分组,属性函数用于控制特殊进程控制函数所用的进程和方法论、控制属性、变量、输入和该特殊函数的输出,以及诸如技术员视图和操作员视图等所需的函数的图形视图。

控制模板系统 120 包括与模板生成器 124 通信的控制模板库 123。控制模板库 123 包含表示各组在进程控制程序中使用的预先定义或已有控制模板函数的数据。一般这些模板与系统一同来自系统设计者,提供给用户。本发明的新模板生成器 124 提供了一允许用户生成新控制模板函数或修改已有控制模板函数的界面,这两种函数也都可存储在控制模板库 123 中。

模板生成器 124 包括属性和方法语言生成器 126 和图形生成器 128。属性和方法语言生成器 124 提供了各种显示屏,允许用户定义多个与生成新控制模板函数或修改某已有控制模板函数有关的属性函数,例如输入、输出和其他属性,而且生成器 124 还提供了能使用户对执行该特定控制模板之新的或经修改的函数的方法或程序进行选择的选择的显示屏。图形生成器 128 提供了用于设计将与特定控制模板相关联的图形视图的方法。正如以下将作进一步讨论的,用户利用属性和方法语言生成器 126 存储的数据和图形生成器 128 来完全确定某控制模板的属性、方法和图形视图。最好将表示已生成的控制模板函数的数据存储存储在控制模板库 123 中,从而以后设计员在设计进程控制解时可以选择和使用它们。

图 5A—G 示出了本发明是如何在一公共进程控制环境中生成图 5A 所示控制模板 130 或修改已有控制模板,以定制一满足开发进程控制函数(例如修改已有的标准进程控制函数或生成专门定制

控制函数有关的图形视图。这样,通过使用图 4 所示的控制模板系统 120 并且遵循此处所示的与图 6—15 有关的指令,可以生成图 5A 所示的控制模板 130。所生成的控制模板包括再次用诸如输入 40 和输出 43 等属性指明的进程控制函数 132。它包括执行该函数所必需的算法 44,参数 46 和 48。算法 44 作用于参数 46 和 48 来利用输入 40 和输出 43 以提供所需进程控制函数。正如下面将更具体描述的,生成多个对话集 42 并使之与进程控制函数 132 相关联。这些对话集通过技术员视图、操作员视图、控制员视图、维修视图和其他所需视图与进程控制函数的参数相联系。图中控制模板视图 34、36 和 38 作为进程控制模板 130 的一部分被生成,并且从技术员视图、操作员视图和实验室视图的立足点出发进行识别。显然,其他视图也可与对话集 42 相关联。然后,可把这些控制模板视图 34、36 和 38 存储在所示的控制模板库 123 中,用户通过界面 118 从库中选择控制模板视图,以在技术员工作站 106 确定控制/监视策略或用预先为操作员定义的模板视图或工作站 102 和 104 中控制/监视策略的实验室界面生成显示。

例如,假设技术员希望在图 5C 所示的输出设备函数 28 的输出端生成一放大函数 34 的技术员视图(幻像显示)。如下所示,他用图 4 所示的模板生成器 124 生成图 5A 所示的模板 130 的技术员视图。同时,他还相应生成与操作员视图、实验室视图、维修视图和其他所需视图有关的对话集 42。因此,如果放大函数 34 要求对技术员视图 118 进行某些电压限制,那么尽管操作员视图上可能没有显示该变化,但实验室视图可能需要显示该变化。然而如下所示,在技术员视图上进行的任何有必要显示在其余视图上的变化将在生成进程控制模板 130 时作出。

图 5B 所示的控制模板库 123 的技术员视图将包括表示新控制模板 130 的数据 34。于是,技术员可以从图 5B 中选择数据 34,以访

问新生成的控制模板 130 的技术员视图,并将它添加到该已定义监视/控制策略的图 5C 中的技术员界面显示 118 上。

用类似的方式,图 5D 所示的控制模板的操作员视图 50 将包括为新控制模板 130 生成的数据 36。于是,操作员可以从图 5D 中的视图选择数据 36,以访问新生成的控制模板 130 的操作员视图,并将它添加到图 5E 的操作员界面显示 118 上,其中图 5E 的操作员界面显示 118 是用图 5D 的操作员视图中其他存储的数据 52、54、58、60 和 62 来构造的。

同样以类似的方式在图 F5 中显示了模板 130 的实验室视图 64。它包括表示新生成的控制模板 130 的实验室视图的数据 38 和其他数据 66 和 68。于是,可用新生成的控制模板 130 的实验室视图 64 来构造图 5G 的实验室界面显示 118。

因此,所生成的进程控制模板 130 形成一公共或普适的环境,例如当通过技术员视图生成或修改后,该环境将相应修改所有其他必要的视图,从而避免为其余每个视图分别生成附加模板。注意在图 5A 中,表示各种视图的控制模板块 34、36 和 38 是模板 130 的一部分,并且存储在控制模板库 123 中,从而可在技术员工作站上生成工作站的界面显示。对于图 4 所示的任何工作站,可通过访问库 123 并指定所需的特定模板函数块来获得任何一种预先定义的视图。现在考虑生成新的控制模板或修改已有控制模板的细节。

参照图 6,显示屏 118 中的视屏显示(“视窗”)200 示出一模板生成器界面,它允许用户与工作站 106 的 CPU116 通信,从而可过与其相关联的函数来定义控制模板。尽管环境 100 中可以存在多个技术员工作站,但在此图中该屏或视窗 200 呈现在工作站 106 的显示屏 118 上。在视窗 200 旁边的是参考块 202 和参考块 204,前者的标题是“下一显示的进度”,示出了可被选择用来表示视窗 200 所需击键顺序的信息,后者的标题是“说明”,示出了可被选择用来说明视窗

200 特定区域的用途的信息。可以理解,除非作出选择,一般参考块 202 和 204 并不作为视窗 200 的一部分或显示屏 118 的一部分出现,但象参考块那样作为对理解本说明书的帮助出现在图 6 和随后的附图中。

模板生成器界面视窗或显示屏 200 包括了像 X—视窗、微软视窗或 OS/2 表示管理器(Presentation Manager)等许多计算机视窗环境通用元件,以下仅作简要描述。视窗 200 在四周包含一边框或框架 206,它确定了视窗相对于显示屏 118 的大小和位置。在模板生成器界面视屏 200 的边框 206 内,可找到许多控制点。其中有用于移动、尺寸和关闭视屏或视窗 200 的控制框 208、用于显示视窗程序名称或标题的标题条 210、在仍然允许程序执行的同时将视屏或视窗 200 缩成一小的图标表示的最小化按钮 212、将模板界面视屏或视窗 200 扩大至整个显示屏大小的最大化按钮 214,以及在视屏或视窗 200 内提供下拉菜单命令的菜单条 216。边框 206 内余下的区域是工作区 218。该区域在用户和控制模板生成器 124 的程序之间提供图形和文本界面,允许用户与 CPU 116 进行通信,生成或修改某选定的控制模板。上述视窗元件在技术上是公知的,故不再进一步讨论。

在工作区 218 中,呈现了打开的记录簿 220 的图形表示。记录簿 220 在用户和模板生成器 124 之间提供一可视界面。该记录簿 220 允许用户阅读和选择代表已有控制模板的数据或将与新生成的控制模板相关联的数据。为了选择允许生成和/或编辑控制模板的显示屏,可用工作站 106 的 CPU 116 选择生成显示屏的数据提示符,显示屏允许生成、执行和监视进程控制程序,并且使用模板生成器 124。记录簿 220 包括一标题部分 222 和一选择部分 224。标题部分 222 为用户提供了当前被选择或打开的某特定项目或记录簿的名称。选择部分 224 提供了一张数据提示符的清单,这些提示符涉及与特殊项目有关的进程,可由用户选择,以生成专门的控制模板或修改

已有控制模板。可以理解,用户用任何常规的视窗的方式“选择”视窗 200 的特殊命令、函数或部分,例如在提示符上双击鼠标器(未示出)或在显示屏 118 一部分显示的诸如键盘等用户接口设备上输入命令键。

在图 6 中,标题块 222 中例示了一题为“汇编者”的被选项目的名称,并且在选择部分 224 中,有关汇编者项目的特殊进程用“域 1”和“域 2”表示。在域 1 和域 2 的每个引用中,有定义为“设备”、“操作”和“控制”等子引用的视屏提示符,设备子引用提示符选择的显示屏涉及有关汇编者项目的实际设备的装置设置,该提示符还选择用于监视设备的诊断界面显示屏。操作子引用提示符所选显示屏涉及已在汇编者项目为域 1 和域 2 引用开发的进程控制程序和相关的操作员界面。控制子引用提示符所选的显示屏涉及与域 1 和域 2 引用相关的控制/监视策略的定义。可将该信息放在控制器/多路复用器中,以提供可供工作站观察的控制或监视函数。用户为了显示以及随后编辑或修改相关屏或开发与所选域的特定方面有关的进程控制程序,可以从一个已定义引用(域 1 或域 2)中选择一子引用提示符。用户选择控制子引用提示符后,便可获得一个提供视屏提示符的控制/监视策略界面显示屏,这些视屏提示符允许生成或修改有关所选子引用的特定进程控制程序的视屏提示符。

在选择区 224 的底部,用视屏提示符显示了“附录”228,包括“费舍库(FISHER LIBRARY)”229 和“用户库”230 的提示符。费舍库视屏提示符 229 选择一系统提供的库,该库包含存储在控制模板库 123 内预先定义或已有的控制模板数据显示屏。这些模板是随系统从设计员那里一同带来的。用户库视屏提示符 230 选择进程控制程序中使用的存储在控制模板库 123 中用户生成的控制模板。通过选择选择区 224 附录中的用户库视屏提示符 230,用户为生成或编辑图 7 所示的已有控制模板 316 可选择第一数据显示屏,以与模板生

成器 124 进行通信。

参考图 7, 视窗 300 相应于第一数据通信视屏, 它包括对已有控制模板数据进行用户库选择的视屏提示符 316 和其他用于其附录 315 中被选项目的系统特性 318、320、322、324、和 326。如图 6 所示, 视窗区 304 中示出了一打开的记录簿 302, 它具有标示了汇编者项目的标题块 306, 还具有有关该项目的标示相同的域 1 和域 2 引用和相应子引用 310、312 和 314 的选择部分 308。但是, 附录 315 的视屏提示符已被改变, 以反映与用户库有关的各种工具或视屏提示符。这些视屏提示符包括控制模板 316、控制规范 318、对话 320、视图 322、趋势 324 和报警 326。附录 315 中的控制模板工具提示符 316 选择存储在控制模板库 122 中的已定义或已有的控制模板。控制规范工具提示符 318 把预先定义的控制单选作显示屏, 用于为被选的进程控制函数定义程序。对话提示符 320 选择一能在工作站和用户之间进行数据通信的可编辑屏, 该屏可与诸如温度、压力之类以下将进一步讨论的特别选定的控制模板属性相关联。视图工具提示符 322 选择与被选进程控制函数之操作环境的大小、颜色等属性有关的用户可编辑的属性。不要将视图工具提示符 322 与有关特定控制模板的图形视图相混淆。趋势工具提示符 324 和报警工具提示符 326 选择被存储起来的用户可编辑的属性, 诸如有关被选进程控制函数的进程趋势和报警(未示出)的颜色、声级等。

用户通过选择选择区 308 中附录 315 的控制模板工具提示符 316, 开始生成一控制模板。如下所述, 一旦选中, 视窗 300 便转换至图 8 所示的控制模板界面视屏, 以允许用户开始定义与特定函数相关的具体属性。

参照图 8, 图中示出了控制模板界面视屏或视窗 400, 它是对图 7 中附录 315 的控制模板工具提示符 316 的选择结果。视窗 400 允许用户定义他将为被选进程控制函数生成的控制模板的类型。在视

窗 400 的区域 420 中示出了第一域或字段:控制域 402,它为用户提供了识别当前视屏提示符的信息,当前视屏提示符选自图 6 所示的附加视屏提示符附录 228 中的用户库 230。界面区 420 还包含用“模板设置”滚动框 404 表示的第二区,“起始模板”滚动框 406 所表示的第三区,“模板名称”字段框 408 所表示的第四区,包含“是”按钮 412a、“消除”按钮 412b 和“帮助”按钮 412c 等三个控制按钮 412 的第五区,和用“模板描述”字段框 410 表示的第六区。滚动框 404 和 406 在框的左边提供了一系列项目,而在框的右边提供了滚动条。用户根据希望的移动方向,通过点击框 404 和 406 所示滚动条的向上或向下箭头来“滚动”被显示的清单。第二区或模板设置滚动框 404 包含一些视屏提示符,它们允许用户定义其希望生成的控制模板的类型。如框 404 所示,用户可以用许多不同的图形进程控制程序设计语言来生成控制模板,设计语言包括连续函数块。梯形逻辑或时序函数图(“SFC”)。应该理解,本发明的进程控制环境 100 内也支持其他的图形进程控制语言。

在图 8 中,假设用户已选择了模板设置滚动框 404 或第二区的“<NEW>CONT”(连续函数块语言)选项,作为他希望生成的控制模板的类型。当用户为新控制模板选择某特定的进程控制语言时,第三区,即起始的模板框 406 中会出现可用来作为设计新控制模板之起始点的预先定义或已有模板的清单。该表与控制模板库 122 中所有涉及模板设置框 404 中被选进程控制语言的预先定义或已有的控制模板有关。如果用户把滚动框 406 中的一个预先定义控制模板选作开始的基础,那么所有与该被选的预先定义模板相关的已有属性、方法和图形视图都变成新控制模板的基础或起始点。如果用户不希望把控制模板库 122 中的预先定义控制模板用作起始点,那么从框 406 中选取提示符“<NEW>”。

一旦用户从滚动框 404 中选取了将生成或修改的控制模板的类

型,并从滚动框 406 中选取了起始的模板,随后便可把新控制模板的专用名称输入到字段框 408 中。输入字段框 408 的名称将是存入控制模板库 122 中的新控制模板的名称,并将形成图 5 所示的以后可以访问该新控制模板的函数块,在该图中,已把“PID—ADP”作为新模板函数块的名称输入。命名新控制模板后,用户可把对新控制模板有意义的描述打字输入“模板描述”字段框 410 中。模板名称和模板描述为从控制单或显示屏程序设计环境中引用新的控制模板提供了机理,并为在进程控制报告和资料中描述该控制模板提供了手段。

在图 8 中,可以看到用户已经根据控制模板 PID 把将被生成的控制模板的类型指定为连续函数块,并且已经提供了识别该控制模板用的专用名称:PID—ADP。如果用户认为他对“模板设置”、“起始模板”和“模板名称”的选择是合适的,那么选择“是”按钮 412a 以存储其定义的参数,然后退出该界面视窗 400。但如果用户希望不存储选择而退出视窗 400,那么选择“取消”按钮 412b。“帮助”按钮 412c 为用户提供了所关联的帮助信息。

参照图 9,图中示出了界面视窗 500 生成特征的选择界面视屏显示。它在用户选择“是”按钮 412a 而退出视窗 400 后提供给用户。在该视窗和随后的视窗中提供了标题条 502,它示出了新控制模板的名称和描述。在该例中,模板名称是“PID—ADP”,而模板描述是“适于 PID”。此时,用户具有能提供生成特征选择的菜单 506,例如通过选择属性定义视窗或视屏来编辑与新控制模板相关的属性,通过选择方法定义视窗或视屏来定义与新控制模板相关的方法,或通过选择视图界面显示视图或视屏来生成与新控制模板相关的视窗。这些生成特征的选择是通过从菜单条 504 选择“编辑”命令,然后从菜单 506 选取的下拉式菜单(未示出)中选择“属性”、“方法”或“视图”等选项来实现的。因此,可在这里设计如图 5 中标号 42 为新的或修改过的控制模板的每个视图所指定的生成特征或对话集。

参照图 10,图中示出了一属性定义视窗或视屏 600,它是在以上述讨论的方式从视窗 500 中选择属性选项后提供给用户的。视窗 600 包含这样一些提示符,它们允许用户定义所有将被包括该新控制模板的特定控制模板使用的属性和有关对话集 42。所有与控制模板相关的和保存在控制模板库 122 中的属性皆在属性视窗 600 内定义。通过在单个界面视窗中提供与全部控制模板有关的所有属性的定义,为控制模板属性建立了一个公共数据库结构,该信息通常将影响进程控制函数的所有视图(技术员的、操作员的等等)。

标题条 620 引用的新控制模板的属性定义显示屏或视窗 600 包括定义属性类型选项的第一区 602、输入属性名称的第二区 604、包含标题为起始属性(ATTRIBUTE TO START FROM)的数据格式视屏提示符的第三区(滚动框 606)、定义有关被选属性之对话集的第四区域(滚动框 608)和包含执行按钮 610a、保存按钮 610b 和删除按钮 610c 等三个控制按钮或视屏提示符 610 的第五区。一个控制模板有下列三种类型的属性组成:输入属性、输出属性和其他属性。属性名称 604 是与控制模板的每个被选属性相关的专用名称。起始属性 606 涉及那些用来开始设计一新控制模板的预先定义的属性,或者与那些在图 8 的起始控制模板滚动框 406 中被选的已有控制模板有关的属性。因此,当在图 8 中选取控制模板 PID 时,有关 PID 的属性(即模式、设置点和状态)作为预先定义属性可以为新控制模板 PID-ADP 采用。这些属性的每个值都可在适当的框中输入。对话集滚动框 608 包含一与滚动框 607 中被选属性有关的所有对话或参数清单。这些对话仍是用来改变新进程控制函数所有视屏显示视图的。

将与新控制模板有关的属性类型选项是在属性类型选择区 602 中定义的。属性类型选择的界面由三个控制按钮组成,可以选择属性类型“输入”、“输出”或“其他属性”。用户通过点击与三个属性类型中

的一个有关的控制按钮,开始定义每个将与新控制模板有关的属性。然后,用户在属性字段 604 内输入属性名称。此例中,用户已在选择区 602 中选择了输出属性类型并在字段 604 中给出了输出属性名称“VO”。应该理解,该属性与标题条 20 中引用的新控制模板 PID-ADP 有关。

在选择了属性类型并为该属性提供了名称之后,用户从第三区 606 选取已命名属性的数据格式或起始属性。被允许的数据格式类型与从图 8 第二区或滚动框 404 的模板集中选取的控制语言类型有关。对于连续控制块语言,数据格式类型包括实数、枚举值、布尔值和正文串。应该理解,滚动框 606 中存在的数据格式类型对于梯形逻辑控制模板可能不同,例如将直接涉及图 8 滚动框 404 内用户可采用的任何进程控制程序设计语言。对已命名属性的数据格式选择在起始属性滚动框 606 内进行。如上所述,用户可从框 606 中选取一种数据格式,或选取框 606 中一种预先定义数据格式。在图 10 中,已定义的数据格式包括模式、设置点和状态。每个预先定义的数据格式为实数、枚举、布尔或串格式中的一种。

当用户从滚动框 606 中选取四种数据类型中的一种时,出现一与该数据格式相关的对话框。为了便于说明,与实数数据格式(<NEW-Real>)相关的对话框示作框 612,与枚举数据格式(<NEW-Enum>)相关的对话框示作 614,与布尔数据格式(<NEW-Bool>)相关的对话框示作 616,而与串数据格式(<NEW-String>)相关的对话框示作 618。因此,当用户选择实数数据格式<NEW-Real>时,出现对话框 612。实数框 612 允许用户填入属性的初始值、属性的下限值和上限值,以及与该属性相关的测量单位。另一方面,如果用户从滚动框 606 中选取了枚举数据格式<NEW-Enum>,则提供框 614,允许用户为该属性至少定义两个枚举值,并用一名称将那些值联系起来。如果用户从滚动框 606 中选取布尔数据格

式<NEW—Bool>,则提供框 616,允许用户定义与零(“0”)和一(“1”)布尔状态有关的名称。最后,如果用户从滚动框 606 中选取串数据格式(<NEW—String>),则提供框 618,允许用户输入诸如有关该属性的名称那样的正文信息。

现在,用户已经在选择区 602 中定义了属性的类型(输入、输出或其他属性),在字段 604 中定义了属性的名称(该图中为“VO”),在框 606 中定义了属性的数据格式,并根据框 606 中被选的数据格式在框 612、614、616 或 618 中定义了与该数据格式有关的具体数据。接着,用户选择将与从对话集滚动框 608 中选取的属性有关的特殊对话。这个对话表(608)包括将被使用的属性的各个方面,它们可按用户的需要进行修改,每个对话集可使用户通过选择引用该属性的显示视图修改属性值。参见图 5A—5G 及其相关的讨论。框 608 中提供的对话表依赖于框 606 中被选数据格式的类型。因此(仅作为举例)某一张对话表适于“实数”数据类型,另一张适于“枚举”数据类型,再一张适于“布尔”数据类型,还有一张适于“串”数据类型。

从框 608 中选定某对话集后,用户可以删除或保存刚刚通过选择适当的按钮 610a、610b 和 610c 而生成的属性。如果用户选取保存按钮 610b,则将存入刚定义的属性,并将接着出现一空白的属性定义视窗 600,允许用户定义将随新控制模板(130)一起选取的另一或第二属性。当用户结束定义所有与新控制模板(130)有关的属性时,选择执行控制按钮 610a,从而保存最后生成的属性并退出属性定义视窗 600。

参照图 11,图中示出了第一方法定义显示屏或视窗 700,它是在用户定义与新控制模板有关的属性结束后,通过从图 9 菜单条 504 的下拉式编辑菜单中选取编辑一方法命令而提供给用户的。控制模板 130 的方法数据形成部分可以以显示序列(display series)700 的形式显示,显示序列 700 包含合适的算法和程序数据,当它们与属性

数据关联时,形成进程控制解。因此,一种“方法”就是一种与执行某进程控制函数或部分函数的控制模板相关的程序。方法可以用 Fortran、C 或任何其他的编程语言来编写。视屏或视窗 700 包括起始目录滚动框 702、函数块或方法名称字段 704、方法描述字段 706、起始方法滚动框 708 和三个控制按钮 710a、710b 和 710c。起始目录框 702 是第一区,它包括一张保存在控制模板库 122 内的所有控制模板的清单。起始方法框 708 是第二区,它提供了一张与选自滚动框 702 的控制模板相关的所有预先定义的方法的清单。一旦选定控制模板,与该被选控制模板关联的方法就可在被定义为滚动框 708 的第二区中被选择。仅作为举例,图 11 中所示的有效起始方法是初始化、改变参数和执行,并适合与如图 8 所选控制模板 PID 相关的预先定义方法。

一旦用户在滚动框 702 中选定控制模板,在滚动框 708 中选定将与新控制模板相关的起始方法,他就必须在方法名称字段或第三区 704 中为该方法提供一个名称。该名称表示图 5 中 34、36、38 和 40 之类的函数块,它定义了方法并且对于有关新控制模板的所有其他方法而言应该是唯一的,以便可以在需要时调用设计好的方法模板。提供了方法名称后,用户可以在方法描述字段或第四区 706 中放入对该已命名的方法的描述。然后,用户通过选择适当的控制按钮 710a、710b 或 710c,用接收或拒绝提示符来保存或删除新方法。

参照图 12,图中示出了第二方法定义视窗或视屏 800,它是在用户选择了图 11 中的“是”按钮或接收提示符 710a 后提供的。第二方法定义视窗或视屏 800 在工作区 802 内示出了一程序设计的文本编辑程序,它允许用户编写和/或编辑与图 11 中第一方法定义视屏所定义的并从其中的框 708 中选取的与已命名的方法相关的程序。作为举例,图 12 示出了从图 11 的滚动框 708 中选作起始方法的初始化方法。可用诸如 C、Fortran 或 UPL 等任何公知的程序设计语言在

该编辑程序中开发程序,并且程序能够使用在其他程序设计环境中生成的被恢复的文本文件。

一旦用文本编辑程序在区 802 中编写了某一程序,用户便可从菜单条 804 中选取保存文件的菜单命令,保存已命名的方法所指定的书写程序。然后,用户可以从菜单条 804 中选择退出文件的命令,以退出方法定义视屏界面 800。如在图 11 中定义已命名的方法和如在图 12 中为定义方法编写程序的进程将继续到开发好所有与新控制模板相关的方法为止。下一个与新控制模板相关的图形视图通过从图 9 的下拉式菜单条 504 和随后的菜单 506 中选取编辑视图的命令来生成。

参照图 13,图中示出了视图定义视窗 900,用于生成与新控制模板相关的视图。视图视窗 900 包括定义起始目录滚动框 902 的第一区、定义起始视图滚动框 904 的第二区、定义视图名称字段 906 的第三区、定义视图描述字段 908 的第四区以及包括诸如“是”按钮 910a、“取消”按钮 910b、和“帮助”按钮 910c 等三个控制按钮的第五区。用类似于先前讨论的定义方法的方式,视图视窗 900 的第一区 902 包含这样一些视屏提示符,它们允许用户在控制模板库 122 内任何已有控制模板的预先定义视图基础上建立视图,或者生成整个新的视图。仅作为举例,第二区或滚动 904 包含用于选择技术员视图、操作员视图或实验室视图的视屏提示符。还可生成维修之类的其他视图。当从图 8 的第三区 406 中将 PID 控制模板选作起始点时,新控制模板 PID-ADP 可采用这些视图。在定义起始视图框 904 的第二区中的技术员、操作员和实验室视图是为被选 PID 控制模板生成的视图。如果需要,可在起始视图框 904 中选取“新”起始视图提示符。如图 13 所示,用户已在第一区或滚动框 902 中选取了 PID-ADP 控制模板,并在第二区或滚动框 904 中将技术员视图选作生成 PID-ADP 控制模板之新技术员视图的起始点。然后,用户在第三

区或视图名称字段 906 中定义新视图的名称,并在第四区或视图描述字段 908 中提供对新视图的描述。一旦确定了新视图的视图名称和描述后,用户可以通过选择适当的控制按钮或视屏提示符 910a、910b 或 910c,用第五区来取消或继续。选择提示符 910a 可显示图 14 所示的视图界面设计视屏,它示出了被生成的技术员视图、操作员视图、实验室视图、控制员视图、维修视图或其他视图。

参照图 14,图中示出了视图界面设计显示视屏或视窗 1000。设计视窗 1000 包括工作区 1002 中的图形编辑程序和设计工具调色板 1004。在 1002 区的图形编辑程序内,根据在图 13 所示视图定义视屏 900 的第二区 904 中选取的起始技术员视图,提供了初始图形视图 1006。此例中,图形视图 1006 为开发新 PID—ADP 控制模板的定制图形视图提供了起始点。在根据预先已有的视图为控制模板生成一定定制图形视图时,图形编辑程序为初始图形视图 1006 提供了与先前在图 10 中定义的输入和输出属性数目相应的、适当数量的输入和输出节点。

初始图形视图 1006 中所示的被选技术员视图包括几个可在通常的技术员视图找到的部分,诸如输入节点 1008、输出节点 1010、符号名称 1012 和标记引用数 1014 等。另外,图形编辑程序 1002 包括几个对通常的技术员视图而言是新的部分,如灵巧字段 1018 等。灵巧字段 1018 可为用户提供有关输入或输出值的信息,或甚至可提供有关输入或输出值的实时趋势数据图。第二图形视图 1006a 表示用户从初始图形视图 1006 中生成的初始图形视图 1006 的定制版本,包括用户添加的定制信息,以新的控制模板。在图 14 中,单个输入属性 1008 和单个输出属性 1010 与 PID—ADP 控制模板的属性定义相对应。对视图 1006 进行图形定义后,用户可以从菜单条执行保存文件的命令(未示出),以保存与其控制模板相关的图形视图。

除了新控制模板的技术员视图以外,如图 15 所示,用户还可以

生成控制模板的其他视图,如操作员视图 1160 等。操作员图形界面视屏 1100 图示包括用于生成与操作员视图相关的图形视图的图形工具调色板 1104 和工作区 1102 中用调色板 1104 生成将与新的或经修改控制模板相关的图形视图的图形编辑程序。图中填料箱 1108 与一调节流入和流出填料箱的流体流量用的流量控制阀 1110 相连。该图形表示是 PID—ADP 控制模板的操作员视图 1106,并且当它被保存时,它将直接与生成或被修改的控制模板相关。用户可以通过在菜单条 1112 的文件菜单项下选择保存(未示出)命令来保存该操作员视图 1106。

参照图 16,图中综述了在本文的说明生成新控制模板 PID—ADP 进程的例子中所提供的信息。综述包括属性定义 1200、方法定义 1202 和图形定义 1204。属性定义包含用户通过图 8—10 中的视屏提示符选取的属性信息。该信息包括属性用途、与控制模板 PID—ADP 相关的属性类型、属性名称、属性范围(如果合适)、属性的默认值(如果合适)和与每个属性相关的特定对话。方法定义 1202 与用户用图 7 和 8 中视屏提示符选取的信息有关。图形定义 1204 与用户用图 13—15 中的视屏提示符选取的信息有关。该信息包括与控制模板 PID—ADP 的技术员视图有关的图形图标 1216,表示控制模板 PID—ADP 的操作员视图的图形图标 1218,和表示控制模板的设备技工视图的图形图标 1219。每个视图可以用数字或通过诸如条状图等图形技术间接地显示被选的属性值。

应该理解,不脱离本发明的精神和范围可在本发明中进行各种变化。例如,可以在诸如 DOSTM或 UNIX[®]等非窗口式的环境中提供图 6—15 中所示的视窗或视屏显示界面。另外,可将用来定义属性、方法的参数和图形定义视窗放在任何界面视屏中,以上所示出的只是为了简明起见。还有,除了连续控制块或梯形逻辑,本发明的控制模板设计环境内还可支持图形进程控制程序设计语言,同时保持模

板生成器的定义结构。

尽管图示和描述了本发明的较佳实施例,但可在所示揭示中作一定程度的修改、变化和替换,并且在某些情况下,将使用本发明的一些特征,而不使用另一些特征。因此,应该广泛地并以与本发明范围一致的方式对后附的权利要求进行解释。

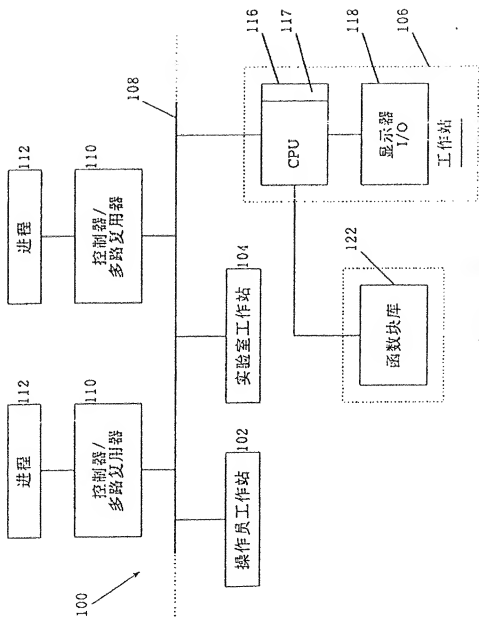
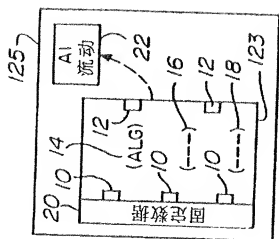
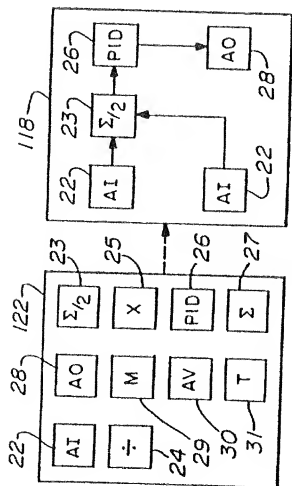


图 1



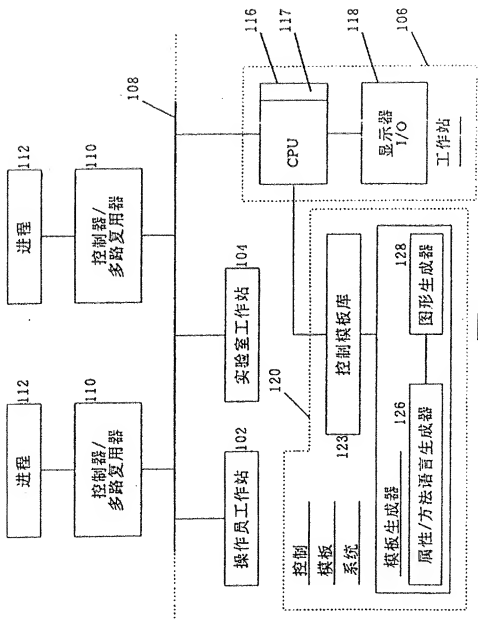


图 4

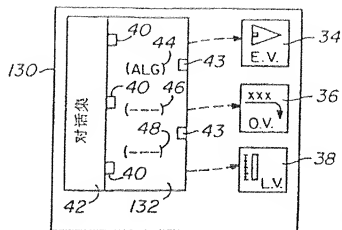


图 5A

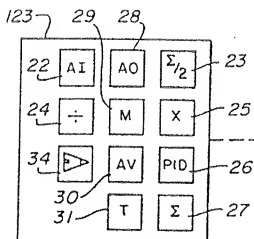


图 5B

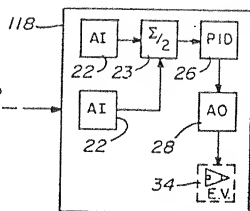


图 5C

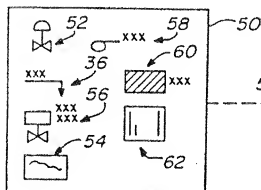


图 5D

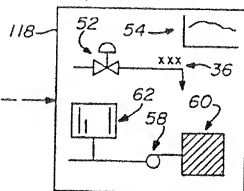


图 5E

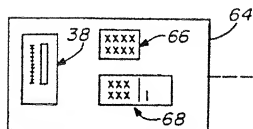


图 5F

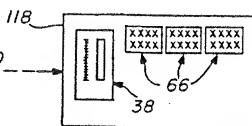
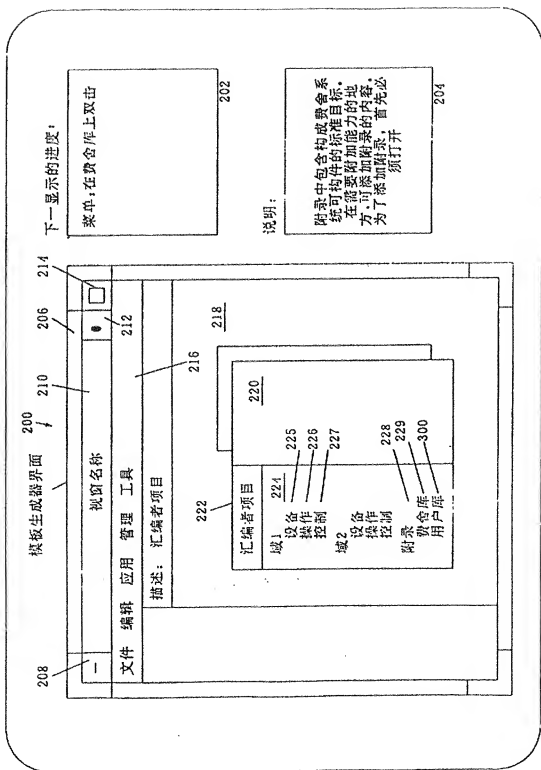
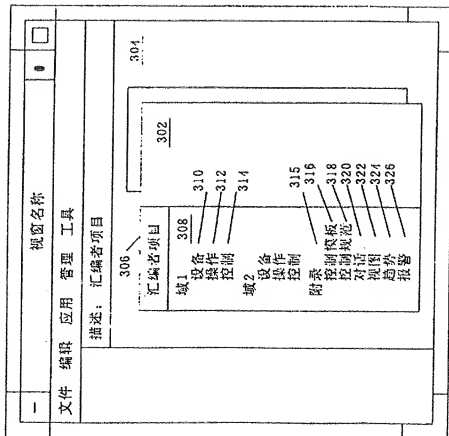


图 5G



300



下一显示的进度:

显示, 在控制模板上双
击

说明:

通过树显示了附录中
信息的分级结构, 对于
控制模板, 用缩进表示
域和子域结构。

图 7

400

下一显示的进度：

显示：点击模板集和起始模板
键输入：输入模板名称和描述
显示：结束所有键输入后选择“是”按钮

说明：

当产生新模板时，可以把一已有的模板用作起始点，当从起始模板中选取<NEW>时，所有与基模板有关的默认数据使用。

视图名称

文件 编辑 应用 管理 工具

控制区

控制模板 402

模板集

起始模板 420

FILTER
P+BIAS
PID
SUMMER
<NEW>
...

模板名称

PID - ADP 408

模板描述

适于PID 410

帮助 412

是 412a

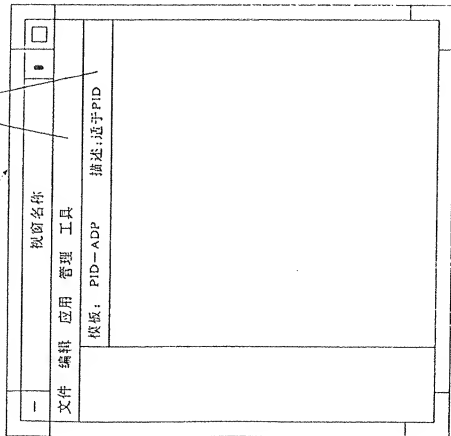
取消 412b

帮助 412c

图 8

500

504 502



下一显示的进度:

菜单: 编辑 → 属性
方法
视图

505

说明:

显示在被修改或定义的
模板名称及其描述,通
过编辑 → 属性首先定义
模板的属性



9

600

视图名称 模板: PID-ADP 描述: 适于 PID 620

文件 编辑 应用 管理 工具

属性类型
☐ 输入
☒ 输出
☐ 其他属性

起始属性
 <NEW-REAL>
 <NEW-ENUM>
 <NEW-BOOL>
 <NEW-STRING>

模式
 设置点
 状态

属性名称
 VO

对话框
 对话框1
 对话框2
 对话框3
 对话框4
 ...

初值
 下限值
 上限值

初值
 名称值
 手动
 自动
 选择
 选择

初值
 名称值
 手动
 自动
 选择
 选择

名称
 开
 关

执行
 存储
 删除

下一显示的进度:

显示: 点击属性类型, 起始属性和对话框, 输入属性名称和所有属性后, 显示: 定义完一个属性后, 点击“存储”按钮, 定义完所有属性后, 点击“执行”按钮。

说明:

从该视图定义模板的所有属性, 当定义完每个属性时, 可将它“保存”入模板定义中, 当定义完所有属性时可选择“执行”按钮退出。

图 10

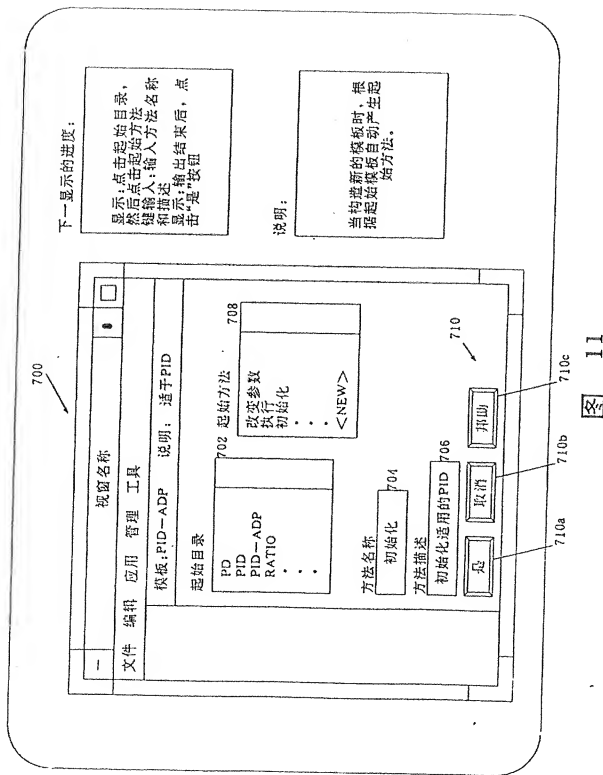
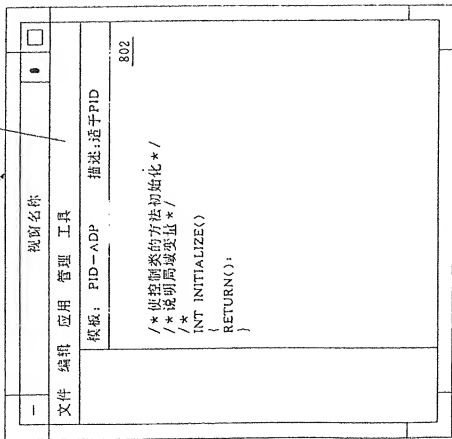


图 11

800

804

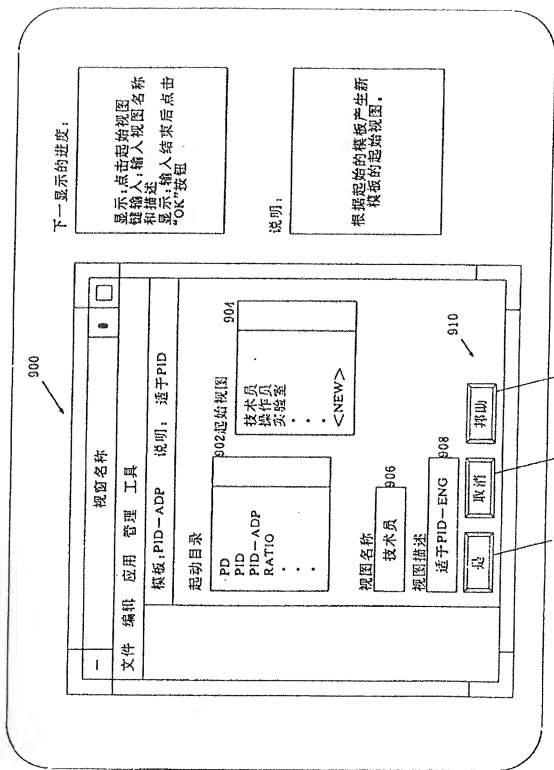


下一显示的进度:

键输入: 为初始化方法
输入UPL或C代码
菜单: 文件 → 新
打开
保存
另存为
关闭
退出

说明:

用文本编辑程序输入或
修改方法的UPL或C代
码,在编辑方法的同时,
可以打开第二视图,允
许方便地访问模板属
性,转换至...



下一显示的进度:

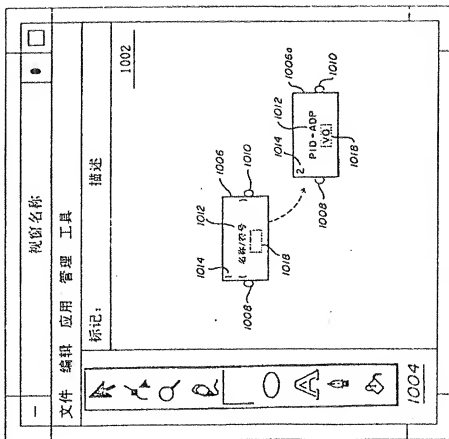
显示: 点击起始视图
按钮输入: 输入视图名称
和描述
显示: 输入结束后点击
“OK”按钮

说明:

根据起始的模板产生新
模板的起始视图。

图 13

1000



下一显示的进度:

键输入: 输入符号和动态信息定义

说明:

根据为模板定义的输入和输出数, 自动提供函数, 用户利用图形编辑程序产生与模板相关的符号和动态信息, 它们将显示在...

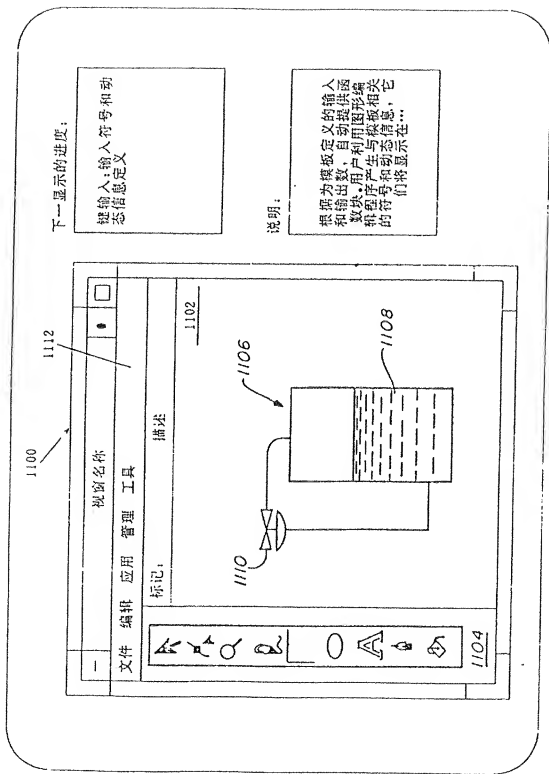


图 15

属性用途	类型	名称	范围	对话
输入	实数	PV	-1E-10至1E10	对话#10
输出	实数	VO	-1E-10至1E10	对话#9
其他属性	枚举	MODE	手动 自动	对话#1
其他属性	实数	SP	-1E-10至1E10	对话#33
其他属性	串	TAG	13字符	对话#22

方法: (过程语言或C代码……按名引用属性)

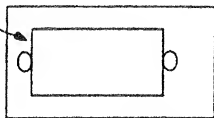
1202

1204

图标

1216

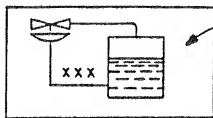
技术员图形



图标

1218

操作员图形



图标

1219

设备人员图形

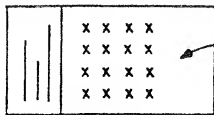


图 16

数信息中包括诸如控制进程的算法和已知控制参数等信息的步骤。

42. 如权利要求 38 的方法,其特征在于,还包括下述步骤,即包括诸如进程输入和/输出等信息作为属性信息。

43. 一种用于在进程控制环境中生成设计控制解时所用控制模板的计算机系统,其特征在于,所述系统包括:

允许用户生成专用控制模板的生成器装置,具体通过对控制属性、用户为处理所述属性所选的控制软件和所述专用控制模板的图形表示进行分组来完成;

用于在用户和所述生成器装置之间提供一可视界面以显示所述专用模板的显示器;

允许用户与所述生成器装置进行通信以改变所述控制属性、所述控制软件或所述专用控制模板的所述图形表示的装置;

用于存储所述专用控制模板,以允许用户选择和调用专用模板,从而在所述显示器上设计控制解的存储装置。

44. 一种用计算机系统在进程控制环境中生成设计控制解时所用控制模板的方法,其特征在于,所述方法包括下述步骤:

允许用户生成专用控制模板,具体通过对控制属性、用户为处理所述属性所选的控制软件和所述专用控制模板的图形表示进行分组来完成;

在用户和所述生成器装置之间提供一可视界面,以显示所述专用模板;

允许用户与所述生成器装置进行通信,以改变所述控制属性、所述控制软件或所述专用控制模板的所述图形表示;

存储所述专用控制模板,以允许用户选择和调用专用模板,从而在所述显示器上设计控制解。